Building Mission Critical Cloud Infrastructure: Lessons Learned At Scale

Eric Westfall

Systems Engineer, DataYard

Who We Are

- Managed service provider specializing in mission-critical cloud infrastructures
- CLEC, blend connectivity with cloud services to produce unique capabilities for our clients
- No commodity services, all we do is five nines
- Small team, big impact. Achieve large scale results through automation and development
- Trusted to architect and host some of the most critical, highest demand applications in our region

Getting On The Same Page

- Asked to define what "the cloud" is, a plurality (29%) of Americans cited some type of weather-related term (e.g., the sky, or an actual cloud).
- When asked whether they believed inclement weather could interfere with cloud computing, 51% of Americans answered yes.
- In this presentation, cloud refers to virtualized infrastructure providing compute, network and storage resources together with agile and resilient network services to provide a robust IaaS platform.

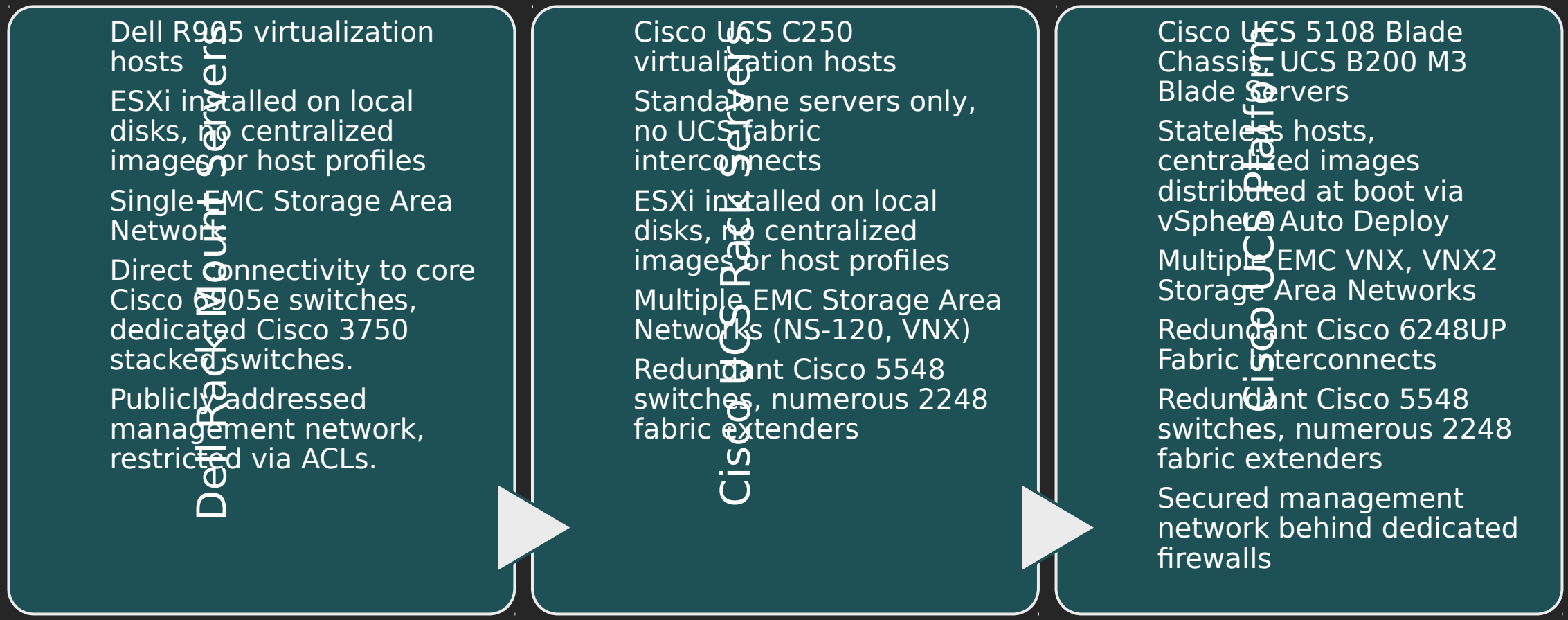
Versatile Infrastructure Platform

Versatile Infrastructure Platform

- DataYard’s Infrastructure as a Service platform trusted to run our critical infrastructure as well as mission-critical platforms for our customers
- Resilient and distributed. No single points of failure.
- Agile networking services. Powerful load balancing, hardware and virtualized firewalls. With layer 2 connectivity, bridge services on internal client networks
- Modular. Easily scale to increased capacity requirements. Additional compute nodes can go from in the box to production in < 60 minutes.
- Standards based. Programmable through vendor CLI/API and custom written APIs.

Platform Evolution

- Complex systems change; our platform has evolved dramatically since initial deployment.
- Flexibility and iteration are key. Don’t get so stuck trying to build the perfect platform that you don’t deploy anything … perfect doesn’t exist.
- The outcome of small measured changes are easier to predict and easier to recover from.
- Our platform has gone through three significant evolutionary phases and many smaller iterations.


Dell Rack Mount Servers
Dell R905 virtualization hosts
ESXi installed on local disks, no centralized images or host profiles
Single EMC Storage Area Network
Direct connectivity to core Cisco 6505e switches, dedicated Cisco 3750 stacked switches.
Publicly addressed management network, restricted via ACLs.
Cisco UCS Rack Servers
Cisco UCS C250 virtualization hosts
Standalone servers only, no UCS fabric interconnects
ESXi installed on local disks, no centralized images or host profiles
Multiple EMC Storage Area Networks (NS-120, VNX)
Redundant Cisco 5548 switches, numerous 2248 fabric extenders
Cisco UCS Converged Infrastructure
Cisco UCS 5108 Blade Chassis, UCS B200 M3 Blade Servers
Stateless hosts, centralized images distributed at boot via vSphere Auto Deploy
Multiple EMC VNX, VNX2 Storage Area Networks
Redundant Cisco 6248UP Fabric Interconnects
Redundant Cisco 5548 switches, numerous 2248 fabric extenders
Secured management network behind dedicated firewalls

Lessons Learned

B:4 S:0xfe31a000600080813
M:0xe00c0ffe010000000 A:0x18284485930 4

Machine Check Exceptions, Memory Errors or How We Learned To Hate The Color Purple

- Platform initially used clustered rack-mount Dell PowerEdge R905 servers (4 Quad-Core AMD Opteron 8356 processors, 128 GB Memory)
- Began experiencing high volumes of single-bit and multi-bit memory errors under heavy workload
- 6 fatal kernel errors (PSOD) in 9 months all precipitated by hardware faults (machine check exceptions in processors, unrecoverable memory errors)
- VMware and Dell agreed root cause was hardware … eventually.
- Agreeing on resolution was not so easy. Replaced two processors, one partial and two complete sets of memory DIMMs, a motherboard and eventually an entire server chassis.

What We Learned

- Some hardware just doesn’t hold up under extremely large or complex workloads. Even when it is the largest platform offered by a vendor.
- Don’t underestimate the ability of your vendors to blame each other. Escalate to the smartest engineers available and then get them on the phone together.
- Even the most thorough hardware diagnostics can fail to uncover issues; some issues can only be discovered under real world workload.
- Admitting is the first step. When you run into a platform limitation, change direction. Don’t succumb to vendor lock-in.

MSCS Clustering (Part 1) - Round Robin Path Selection and RDM LUNs

- Default path selection behavior favors interface failover not load balanced I/O performance.
- Troubleshooting storage performance in these environments is complex enough – in some configurations, fixed path selection can result in random path changes after reboots further complicating troubleshooting.
- Huge I/O performance gains when using round robin path selection but can cause issues in Microsoft Clustering environments.
- Prior to vSphere 5.5, round robin path selection with MSCS was not supported and would break shared storage when LUNs were mapped as RDMs.

What We Learned

- Path selection policy decisions should be made at individual LUN levels and not simply applied to all LUNs
- Microsoft clustering using native iSCSI and LUNs mapped as RDMs is just awful in vSphere versions prior to 5.5 … more on that later
- Pay attention to graphs and performance metrics, active/passive failover is nice but redundancy *and* performance gains are even better.

MSCS Clustering (Part 2) – Improved Boot Performance With Perennial Reservations

- MSCS performs storage arbitration using SCSI-3 reservations
- The vSphere storage subsystem attempts to discover all devices presented to an ESXi host during the device claiming phase
- MSCS RDM LUNs with a reservation placed on them from an active MSCS node hosted on another ESXi host prevent the booting host from interrogating the LUN.
- Use the supported flag to mark RDM LUNs participating in MSCS clusters as perennially reserved so the storage subsystem skips LUN interrogation during device claiming
- 83% host boot time reduction on average (41 minutes -> 6.5 minutes)

What We Learned

- Did I mention MSCS using native iSCSI and LUNs mapped as RDMs sucks … cause it does.
- Using in-guest iSCSI software initiators with MPIO is a much better shared storage alternative to native RDM LUNs and reduces overall complexity
- Don’t ignore performance issues or assume long boot times are normal just because these are big servers with tons of memory or a lot of LUNs to discover.

Fabric Extender Buffering, Queue Limits and Tail Drops

- The Cisco Nexus 2248TP fabric extender uses a shared packet buffering scheme where 8 host interfaces (HIF) map to a single ASIC with 800 KB N2H; 480 KB H2N.
- Buffers are needed where speed mismatch occurs, as in all network designs and in particular when the bandwidth shifts from 10 GB to 1 GB (N2H).
- If the host interface is congested, traffic is dropped according to the normal tail-drop behavior.
- Default queue tail-drop threshold of 64 KB N2H, can be removed to allow each HIF to access full shared memory buffer (dependent on number of NIFs configured).

What We Learned

- Pay close attention to the specifications of your switching fabric, dig deep into architectural details and capabilities.
- Block storage traffic is bursty and doesn’t play well in limited shared packet buffering architectures. Make sure you have a large enough shared buffer to deal bursty traffic and speed changes.
- Cisco now manufactures specialized fabric extenders (i.e. 2248TP-E) optimized for big-data deployments and distributed storage. 32 MB shared buffer space, not dependent on the number of NIFs, default queue limit 1 MB H2N.

Distributed Virtual Switch Maximum Heap Allocation

- Issues running distributed virtual switches at large scale deployments; dropped virtual machine network connectivity, errors when powering on virtual machines.
- Errors in vmkernel log: “Failed to get DVS state from vmkernel Status (bad0014)= Out of memory”; “Unable to Add Port; Status(bad0006)= Limit exceeded”; “WARNING: Net: vm 735381: 4454: cannot enable port 0x4000037: Out of memory”
- Resolved by increasing the large heap maximum allocation size for the distributed virtual switch.
- Was a “non-public” bug, now publicly disclosed (2034073).

What We Learned

- Vendors (especially VMware) withhold bugs from public disclosure … lots of them. Maintain partnerships and support contracts since you can’t always guarantee your issue is on the knowledgebase.
- Centralized logging from your hosts is crucial; review vmkernel logs for obscure bugs and track down abnormal errors
- For some issues, there just isn’t a best practice recommendation available. VMware still does not publish recommended port maximums as they relate to heap values. Official recommendation is to contact support if you reach the maximum heap value of 128 and still have issues.

Final Thoughts

- Things break, unexpectedly … focus on mean time to recovery not mean time between failure
- Distributed systems are inherently complex; favor simplicity wherever you can find it.
- Eat your own dog food, build a platform you trust to run your critical infrastructure. And hey, if you’re building it for yourself … why not sell it?
- Iteration, iteration and more iteration. What you build will change, I guarantee it. Embrace change, incorporate lessons learned and continuously improve the platform.

Questions?

More info:

Eric Westfall

eric.westfall@datayardworks.com

(800) 982-4539

http://datayardworks.com